Программа

Введение

1. Примеры наиболее дорогих ошибок
2. Причины возникновения:
    - Космические лучи :)
    - Ошибки в ПО
3. Виды тестирования (различные классификации)
    - По цели
    - По свойствам
    - По исполнителю
    - По уровню
    - По интерфейсу
4. Политика версионирования при тестировании:
    - Альфа-версии
    - Бета-версии

Выбор вида тестирования

1. Пирамида тестирования:
    - Приоритеты различных видов тестирования
    - Соотношение видов тестов
2. Принципы семантического версионирования
    - Стандарт версионирования
    - Пример на разделяемом протоколе взаимодействия
    - Пример на библиотечных решениях

Тестирование API. WSDL

Теория

1. Особенности тестирования протоколов
    - RPC: WSDL/SOAP + REST/JSON
    - Messaging
2. XSD-схемы - основы и примеры описания типов
3. WSDL-сервис - основы и пример описания методов
4. Пример генерированного кода для сервера и клиента

Практика

1. Сетевой сниферинг SOAP с помощью wireshark
2. SoapUI:
    - Создание проекта на основе WSDL

- Посылка запросов и получение ответов
- Создание TestSuit
- Проверки на основе XPath

Планирование тестирования. Test case / Bugs

1. Структура и назначение Test-plan:
   - Кто
   - Что
   - Как
   - Когда
   - Критерии
2. Структура и назначение Test-case:
   - Предусловия
   - Шаги
   - Фокусирование на функциональности
3. Заведение ошибок:
   - Workflow
   - Основные поля и принципы их заполнения

Тестирование API. REST

Теория

1. Напоминание принципов протокола HTTP (GET/POST/…)
2. JSON-schema/OpenAPI/Swagger - основы и примеры описания REST API
3. Аналогия с XSD/WSDL

Практика

1. Postman:
   - Импорт описания API
   - Применение окружений
   - Создание запросов
   - Использование переменных и их переопределение
   - Тесты на JS для проверки:
     1. Кода возврата
     2. Полей ответа
     3. Соответствия схеме
   - Назначение и применение mock-серверов

Тестирование интерфейса пользователя. Web

Теория

1. Архитектура Selenium:
    - WebDriver
    - API на Python, Java, ...
    - IDE как расширение браузеров
2. Принципы идентификации элементов web-страниц

Практика

1. Selenuim:
    - Создание виртуального окружения на Python
    - Запуск WebDriver
    - Поиск элементов на странице (css, id, атрибуты)
    - Ввод текстовых данных
    - Автоматизированная генерация сценария в IDE

Нагрузочное тестирование

Теория

1. Фокусы нагрузочного тестирования:
    - Производительность
    - Стабильность
    - Отказоустойчивость
    - Масштабируемость
    - Стресс-тестирование
2. Профили нагрузки:
    - SLA
    - Пределы производительности
3. Параметры:
    - Время обработки
    - Чатота запросов
    - Размер данных
4. Откуда брать профили нагрузки:
    - БД
    - Журналы
    - Прогноз
5. Инструменты:
    - Web-консоль
    - JMeter
    - Gatling
    - K6

Практика

1. JMeter:

- Поддерживаемые протоколы
- Ручное создание HTTP-запросов
- Запись сценариев через Proxy
- Thread group и его параметры
- Вынесение общих параметров
- Просмотр результатов в графическом и табличном видах

Тестирование интерфейса пользователя (Desktop)

Теория

1. Примеры технологий разработки интерфейса и соответствие инструментов тестирования со знанием идентификаторов элементов интерфейса:
   - Qt: Squish
   - JS: Selenium
2. Применение компьютерного зрения: Sikuli
   - Архитектура
   - OpenCV
   - Tesseract
   - Jython
3. Применение машинного обучения: Testolang
   - Архитектура
   - QEMU/KVM
   - Нейронные сети

Практика

1. Sikuli:
   - Подключение sikuli в Python
   - Тест сложения в калькуляторе передачей нажатия клавиш
   - Фиксация изображений для поиска
   - Параметры поиска изображения

Исследовательское тестирование

1. Test strategy model:
   - Function
   - Claims
   - Domain
   - User
   - Stress
   - Risk
   - Flow
   - Automatic
   - Scenario
2. Заведение ошибок:
   - Поля

- Поиск дубликатов по стекам

Fuzzing-тестирование

Теория

1. Виды верификации:
   - Статическая
   - Динамическая (..., fuzzing, ...)
2. Sanitizers:
   - asan
   - ubsan
3. Генерация данных:
   - Начальная выборка
   - Контроль трасс исполнения
   - Эволюционные алгоритмы
4. Критерии остановки тестирования

Практика

1. AFL fuzzer:
   - Сборка clang с ключами asan и ubsan
   - Создание тестовых данных для затравки
   - Пример на дереве условных операторов
   - Запуск afl-fuzz и разъяснение полей, выводимых в runtime
   - Разбор результата поска падения приложения

Классификация методов test design

1. Black Box
   - Классы эквивалентности
   - Граничные значения
   - Доменный анализ
   - Диаграмма переходов состояний
   - Попарное тестирование
   - Тестирование вариантов использования
2. White Box
   - Потоки управления
   - Потоки данных
3. Experience based
   - Checklists
   - Исследовательское тестирование